

CManage Plugins

CManage Registry supports several types of Plugins in order to easily customize and extend Registry functionality. In addition, you can write your own plugins to customize Registry to work in the ways and with the systems that you need. This page provides background about CManage Registry plugins including how to make them available to your instance of CManage, a description of each of the plugin types with a list of the known plugins of each type, and references to documentation to create your own plugins.

On this Page

- 1. About Plugins
 - 1.1. Installing Plugins
 - 1.1.1. Core Plugins
 - 1.1.2. Supported Non-core Plugins
 - 1.1.3. External Plugins
 - 1.2. Configuring & Using Plugins
 - 1.3. Creating Your Own Plugins
 - 1.3.1. Start Here
 - 1.3.2. Additional Requirements by Plugin Type
- 2. Plugin Library
 - 2.1.1. Authenticator Plugins
 - 2.1.2. Cluster Plugins
 - 2.1.3. Dashboard Widget Plugins
 - 2.1.4. Data Filter Plugins
 - 2.1.5. Enrollment Flow Plugins
 - 2.1.6. Identifier Validator Plugins
 - 2.1.7. Invitation Confirmer Plugins
 - 2.1.8. Job Plugins
 - 2.1.9. LDAP Schema Plugins
 - 2.1.10. Normalization Plugins
 - 2.1.11. Organizational Identity Source Plugins
 - 2.1.12. Provisioner Plugins
 - 2.1.13. Vetting Plugins

1. About Plugins

1.1. Installing Plugins

There are a few different ways to make Plugins available for use within Registry, according to how the Plugin is distributed.

1. **Core Plugins** - Commonly used. These plugins ship with CManage and enabled by default.
2. **Supported Non-core Plugins** - Less widely used. Shipped with CManage, but not enabled by default.
3. **External Plugins** - Finally, Plugins can come from other sources (including those you write yourself) to support your needs.

1.1.1. Core Plugins

Many Plugins are already set up, and provided as "Core Plugins"; these Plugins ship with CManage, and are enabled by default. They can be found in the `app/Plugin` directory.

1.1.2. Supported Non-core Plugins

In addition, there are some Plugins that are less widely used. While they are shipped with CManage, they are not enabled by default because of the related overhead (and, in some cases, external dependencies) associated with Plugins that may only be useful to some deployments. These plugins are also shipped with Registry and can be found in the `app/AvailablePlugin` directory.

2. Plugin Library

Below is a list of known plugins that are available to the CManage Registry community. Please let us know if you know of other plugins that are not listed below. (Or, maybe you have created your own plugin that you want to share here!)

2.1.1. Authenticator Plugins

- [Certificate Authenticator Plugin](#) — The Certificate Authenticator plugin manages information about X.509 Certificates for CO People. (*experimental*)
- [Password Authenticator Plugin](#) — The Password Authenticator plugin manages passwords for CO People. (*experimental*)
- [Privacy IDEA Authenticator Plugin](#) — The Privacy IDEA Authenticator plugin provides an interface for tokens managed by a [privacyIDEA](#) authentication server. These tokens can be used to implement Multi-Factor Authentication (though instructions for doing so are beyond the scope of this document). (*experimental*)
- [SSH Key Authenticator Plugin](#) — The SSH Key Authenticator plugin manages SSH Public Keys for CO People.

2.1.2. Cluster Plugins

- [Unix Cluster Plugin](#) — The Unix Cluster plugin manages Unix Accounts for CO People.

2.1.3. Dashboard Widget Plugins

The simplest way to enable a Non-Core Plugin (assuming you have met any external dependencies the Plugin may require) is to symlink it from your `local/Plugin` directory.

After creating the symlink, you should clear caches and then update your database schema to reflect the newly available plugin.

Enabling a Non-core Plugin

```
$ cd $REGISTRY/local/Plugin
$ ln -s ../../app/AvailablePlugin/SomePlugin .
$ cd $REGISTRY/app
$ su -c "./Console/clearcache" ${APACHE_USER}
$ su -c "./Console/cake database" ${APACHE_USER}
```

1.1.3. External Plugins

Plugins that come from other sources (including those you write yourself) should be placed in the `local/Plugin` directory.

1.2. Configuring & Using Plugins

Once a Plugin is installed and enabled for use, how it is actually used and/or configured varies according to the Plugin type.

| Plugin Type | Description | Where the plugin is configured in CManage |
|----------------------------------|---|---|
| Authenticator | Authenticators are used to prove a CO Person's identity to an application or service. Because Authenticators are collaboration issued, they are attached to the CO Person, not to Organizational Identities (as for external credentials). | |
| Cluster | Clusters are used to represent a CO Person's accounts within a given application or service. The typical use case is managing accounts on (eg) one or more Unix servers, but there is no specific requirement for this type of service. | |
| Dashboard Widget | Registry v3.2.0 introduces <i>Dashboards</i> , simple information portals. Information is provided by <i>Dashboard Widgets</i> , which are implemented as <i>Dashboard Widget Plugins</i> . Multiple Dashboards can be defined within a CO. Dashboards include header and footer text areas appropriate for providing arbitrary html content on the landing page of a CO. | |
| Data Filter | | |
| Enrollment Flow | | |
| Identifier Validation | | |
| Invitation Confirmer | | |
| Job | | |
| LDAP Schema | | |

- [Announcements Dashboard Widget](#) — The Announcements Dashboard Widget (AnnouncementsWidget) is a [Registry Dashboard](#) plugin that manages and renders Announcements.
- [Email Address Dashboard Widget](#) — The Email Address Dashboard Widget is a [Registry Dashboard](#) plugin that allows for self service management of Email Addresses.
- [Notifications Dashboard Widget](#) — The Notifications Dashboard Widget (NotificationsWidget) is a [Registry Dashboard](#) plugin that renders [Registry Notifications](#) for the current user.
- [Password Dashboard Widget](#) — The Password Dashboard Widget is a [Registry Dashboard](#) plugin that allows for self service management of Passwords by authenticated users.
- [Recovery Dashboard Widget](#) — The Recovery Dashboard Widget (RecoveryWidget) is a [Registry Dashboard](#) plugin that enables self service account recovery tools.
- [Services Widget Plugin](#) — The Services Dashboard Widget (ServicesWidget) is a [Registry Dashboard](#) plugin that renders [Registry Services](#) for the current user. At this time, the services widget supports only CO-level services.
- [URL Dashboard Widget](#) — The URL Dashboard Widget (UrlWidget) is a [Registry Dashboard](#) plugin that renders content from URLs.

2.1.4. Data Filter Plugins

- [Data Scrubber Filter Plugin](#) — The Data Scrubber Filter Plugin filters which attributes are passed from an [Organizational Identity Source](#).
- [Elector Data Filter Plugin](#) — The Elector Data Filter evaluates available attributes and elects one based on configured Precedence Rules. The elected value is passed to the associated Provisioning Target.
- [Group Filter Plugin](#) — The Group Filter Plugin filters which CO Groups are passed through to the connected context. Rules are processed for each CO Group passed to the Group Filter Plugin.
- [Group Name Filter Plugin](#) — The Group Name Filter Plugin replaces a CO Group Name with an Identifier, so that the name of the CO Group appears to be the Identifier.

2.1.5. Enrollment Flow Plugins

- [DuplicateCheckEnroller Plugin](#) — The Duplicate Check Enroller Plugin checks whether the enrollee has been registered in the past thus preventing Enrolling again and creating duplicates. The check takes place after the start step of the Enrollment Flow. This means that the Enrollment flow should have an [Intro duction text configured](#).
- [IdentifierEnroller Plugin](#) — The Identifier Enroller Plugin allows the enrollee to select one or more identifiers as part of the enrollment process. This happens after email confirmation (and so after the initial petitioner attributes are collected).
- [MeemEnroller Plugin](#) — MEEM is the *MFA Enrollment and Exemption Manager*. It is intended to coordinate enrollment in Multi-Factor Authentication. MEEM does not work with any specific technology, but is intended to work with Enrollment Flows and, indirectly, [Authenticators](#).
- [NationalityEnroller Plugin](#) — The NationalityEnroller allows a Petitioner to self assert their national affiliation(s), which are then stored as [Identity Documents](#), with a Document Type of *Self Assertion*. This plugin supports [Attribute Enumerations](#) for the attribute *Identity Document (Issuing Authority, Self Assertion)*.
- [ServiceEligibilityEnroller Plugin](#) — The Service Eligibility Enroller Plugin allows for [Registry Services](#) to be selected for a CO Person Role. Despite its name, the Service Eligibility Enroller can be used without Enrollment Flows.

2.1.6. Identifier Validator Plugins

- [Dictionary Identifier Validator Plugin](#) — The Dictionary [Identifier Validator](#) determines if an Identifier (manually or [automatically assigned](#)) matches a [Dictionary Entry](#).
- [LDAP Identifier Validator Plugin](#) — The LDAP [Identifier Validator](#) queries an LDAP server to determine if an Identifier (manually or [automatically assigned](#)) is in use (and therefore not permitted).

| | | |
|--|--|--|
| Normaliza tion | | |
| Organiza tional Identity Source | | |
| Provision er | | |

1.3. Creating Your Own Plugins

Building your own plugin is a great way to get started with custom development for COnanage. Through plugins you can build powerful extensions to the existing Registry functionality. Building a Registry Plugin requires knowledge of PHP, CakePHP, and COnanage.

1.3.1. Start Here

The general documentation for building plugins can be found on the [Writing Registry Plugins](#) page. All plugins follow some standard conventions which are outlined on this page. In addition, some types have additional requirements as listed below.

1.3.2. Additional Requirements by Plugin Type

As described above, there are several different Plugin Types, each with their own additional requirements. You can find these details at the links below:

- [Authenticator Plugins](#)
- [Cluster Plugins](#)
- [Dashboard Widget Plugins](#)
- [Data Filter Plugins](#)
- [Enrollment Flow Plugins](#)
- [Identifier Validation Plugins](#)
- [Invitation Confirmer Plugins](#)
- [Job Plugins](#)
- [LDAP Schema Plugins](#)
- [Normalization Plugins](#)
- [Organizational Identity Source Plugins](#)
- [Provisioner Plugins](#)

- [Regex Identifier Validator Plugin](#) — The [Regex Identifier Validator](#) determines if an Identifier (manually or [automatically assigned](#)) conforms to an acceptable regular expression.

2.1.7. Invitation Confirmer Plugins

Content by label

There is no content with the specified labels

2.1.8. Job Plugins

- [Core Job Plugin](#) — The Core Job Plugin provides several out-of-the-box jobs for use with the [Registry Job Shell](#).

2.1.9. LDAP Schema Plugins

Content by label

There is no content with the specified labels

2.1.10. Normalization Plugins

Content by label

There is no content with the specified labels

2.1.11. Organizational Identity Source Plugins

- [API Source](#) — The API Organizational Identity Source Plugin is designed to integrate using RESTful APIs and message buses.
- [Env Source](#) — The Env Organizational Identity Source Plugin is designed to pull attributes from environment variables, generally populated by web server authentication modules.
- [File Source](#) — The File Organizational Identity Source Plugin is designed to integrate using CSV files.
- [LDAP Source](#) — The LDAP Organizational Identity Source Plugin is designed to integrate with an LDAP server.
- [netFORUM Source](#) — The netFORUM Organizational Identity Source Plugin is designed to integrate with the [Abila netFORUM Association Management System](#) via the XML API (xWeb).
- [Novi Source](#) — The Novi Organizational Identity Source Plugin is designed to integrate with the [Novi Association Management Software](#) platform using the [Novi AMS API](#).
- [ORCID Source](#) — The ORCID Organizational Identity Source Plugin is designed to integrate with the [ORCID API](#) in order to create Organizational Identities and to securely link an ORCID iD to an existing CO Person Record.
- [Salesforce Source](#) — The Salesforce Organizational Identity Source Plugin is designed to integrate with [Salesforce](#) via the [Force.com REST API](#).
- [SQL Source](#) — The SQL Organizational Identity Source Plugin is designed to integrate inbound data via a SQL database. SqlSource is available as of Registry v4.1.0.

2.1.12. Provisioner Plugins

- [API Provisioning Plugin](#) — The API Provisioning Plugin provisions CO Person records to a RESTful or messaging endpoint.
- [Changelog Provisioning Plugin](#) — The Changelog Provisioning Plugin is a simple plugin that generates logfile entries on provisioning events. Entries are JSON encoded representations of CO Person and CO Group data.
- [Crowd Provisioning Plugin](#) — The Crowd Provisioning Plugin provisions CO Person and CO Group records to [Atlassian Crowd](#).
- [GitHub Provisioning Plugin](#) — The GitHub Provisioning Plugin synchronizes Registry data with [GitHub](#).
- [Grouper Provisioning Plugin](#) — The Grouper Provisioning Plugin provisions groups and memberships in groups to an Internet2 Grouper instance using the Grouper web services interface.
- [Homedir Provisioning Plugin](#) — The Homedir Provisioning Plugin is an experimental plugin that creates Unix home directories entries on provisioning events. It is not intended for use in a production environment. (*experimental*)

- [Jira Provisioning Plugin](#) — Registry v4.0.0 introduces the Jira Provisioning Plugin, which provisions CO Person and CO Group records to [Atlassian Jira](#).
- [LDAP Provisioning Plugin](#) — The LDAP Provisioning Plugin is designed to provision Registry data into an LDAP server.
- [Mailman Provisioning Plugin](#) — The Mailman Provisioning Plugin manages [Mailman3](#) mailing lists using Registry data. (*experimental*)
- [MediaWiki Provisioning Plugin](#) — The MediaWiki Provisioning Plugin provisions Registry data to a [MediaWiki](#) instance deployed with the [OAuth extension](#). Since MediaWiki is not designed for group based authorization the plugin does not provision group information or memberships to MediaWiki. (*experimental*)
- [MidPoint Provisioning Plugin](#) — The MidPoint Provisioning Plugin provisions users to [Evolveum midPoint](#) using the [midPoint REST API](#). (*experimental*)
- [Salesforce Provisioning Plugin](#) — The Salesforce Provisioning Plugin provisions Contacts to [Salesforce](#) via the [Force.com REST API](#).
- [SQL Provisioning Plugin](#) — The SQL Provisioning Plugin provisions CO Person and CO Group records to a SQL database.

2.1.13. Vetting Plugins

- [Visual Compliance Vetter Plugin](#) (CManage)
- [Dictionary Vetter Plugin](#) (CManage)